

# Interfaces et collections

## LIST

### Ecrivez la classe Etudiant

Un étudiant est caractérisé par un id, un nom et un prenom. Cette classe possède :

- 2 constructeurs dont un sans paramètre.
- les méthodes **getId()**, **getNom()**, **getPrenom()**, **setId(..)**, **setNom(...)**, **setPrenom(...)** permettant d'accéder aux attributs en écriture et en lecture.
- la redéfinition de la méthode méthode *boolean equals(Object o)*, permettant de tester l'égalité entre 2 étudiants selon l'id.
- la redéfinition de la méthode *toString()*, permettant de donner une représentation de l'étudiant sous forme d'une chaîne de caractères

Ecrire un programme main permettant de tester toutes les méthodes de la classe.

### Créez l'interface University

```
public interface University{  
    public void ajouterEtudiant(Etudiant e);  
    public boolean rechercherEtudiant(Etudiant e);  
    public boolean rechercherEtudiant(String nom);  
    public void supprimerEtudiant(Etudiant e);  
    public void displayEtudiants();  
    public void trierEtudiantsParId();  
    public void trierEtudiantsParNom();  
}
```

### Créez les classes EspritVector et EspritArrayList

Construisez les classes EspritVector et EspritArrayList qui implémentent l'interface University et qui reposent sur l'utilisation respective d'un Vector et d'un ArrayList pour la gestion de la liste des étudiants.